

Listing of Claims

29. (new) A method for processing data objects in a distributed data processing system, said distributed data processing system having a plurality of software and/or hardware nodes being communicatively connectable, the method comprising the steps of:

defining a first object processing environment for processing objects at a first level of abstraction, said first object processing environment comprising a first object model with a first category of object aspects, wherein the processing of said first object processing environment is independent of the software/hardware platforms of said nodes;

defining a second object processing environment for processing objects at a second level of abstraction, said second object processing environment comprising a second object model with a second category of object aspects, wherein the processing of said second object processing environment is dependent on the software/hardware platform of at least one of said nodes;

synchronizing said first object model of said first object processing environment with said second object model of said second object processing environment, such that data coherency is maintained in said first and second object models;

defining an object in accordance with said first and second models and for said object associating a selectable set of object aspects from said first and second object aspect categories;

processing an instance of such an object according to said first and second object models in said first and second object processing environments dependent on said associated set of object aspects.

30. (new) The method as recited in claim 29, wherein predefined object connectivity means, in the shape of software code portions and protocols communicatively coupled to said first, platform independent environment, are devised to interface with and execute

data communications between an internal entity situated within and an external entity situated outside said defined first and second environments.

31. (new) The method as recited in claim 29, wherein an instance of said first, platform independent environment is configured together with an instance of said second, platform dependent environment and object connectivity means to constitute an object provider functionality.

32. (new) The method as recited in claim 29, wherein an instance of said first, platform independent environment is configured together with an instance of said second, platform dependent environment to constitute an object consumer functionality.

33. (new) The method as recited in claim 32, wherein an object provider functionality is configured together with an object consumer functionality constituting a consumer/provider subset of the distributed system.

34. (new) The method as recited in claim 33, wherein routing means enables routing of data and control signals within said consumer/provider subset of the distributed system.

35. (new) The method as recited in claim 34, wherein authentication means provides authentication functionality in accessing said consumer/provider subset of the distributed system.

36. (new) The method as recited in claim 35, wherein provisioning means enables provisioning of objects to a consumer within said consumer/provider subset of the distributed system and adapts said objects to the technical configuration of the consumer apparatus.

37. (new) The method as recited in claim 29, wherein said object instance when it is generated is associated with an identification indication that is unique within said distributed system.

38. (new) The method as recited in claim 29, wherein an object is defined by means of a plurality of aspects comprising an object definition, an interface definition, an object instance aspect and an object instance reference aspect.

39. (new) The method as recited in claim 29, wherein data is encapsulated in an object instance by means of an attribute devised to store a selectable predetermined type of basic entity for containing data, said type of basic entity being dependent on the type of data to be stored.

40. (new) The method as recited in claim 39, wherein an attribute of a first object instance is devised to contain a reference to a second object instance for accessing or activating a selectable content of said second object instance.

41. (new) The method as recited in claim 29, wherein an object instance is accessed or activated by means of a portion of executable code associated with said object and devised to perform a predefined task of said object.

42. (new) The method as recited in claim 41, wherein said portion of executable software code is available and devised to activate execution of functionality or operations of said object instance, or to access or manipulate data that is encapsulated in the object.

43. (new) The method as recited in claim 29, wherein information is encapsulated in an object instance at a first node in said distributed system and is accessible, dependent on

predetermined access parameters, from an arbitrary second node in said distributed system.

44. (new) The method as recited in claim 29, wherein a service, possibly involving software logic, is encapsulated in an object instance at a first node in said distributed system and is accessible, dependent on predetermined access parameters, from an arbitrary second node in said distributed system.

45. (new) The method as recited in claim 29, wherein an object is devised to represent a physical entity, an object instance thus being defined to access, interface with or control said physical entity.

46. (new) The method as recited in claim 29, wherein all information within the distributed system is stored in a common and single format and structured in a common way independent of the type of said information.

47. (new) The method as recited in claim 29, wherein an object and a software component within said distributed system are expressed in a common predetermined language and with a common predetermined set of language rules.

48. (new) A system for processing data objects in a distributed data processing system, said distributed data processing system having a plurality of software and/or hardware nodes being communicatively connectable, comprising means for realizing the steps of:

defining a first object processing environment for processing objects at a first level of abstraction, said first object processing environment comprising a first object model with a first category of object aspects, wherein the processing of said first object processing environment is independent of the software/hardware platforms of said nodes;

defining a second object processing environment for processing objects at a second level of abstraction, said second object processing environment comprising a second object model with a second category of object aspects, wherein the processing of said second object processing environment is dependent on the software/hardware platform of at least one of said nodes;

synchronizing said first object model of said first object processing environment with said second object model of said second object processing environment, such that data coherency is maintained in said first and second object models;

defining an object in accordance with said first and second models and for said object associating a selectable set of object aspects from said first and second object aspect categories;

processing an instance of such an object according to said first and second object models in said first and second object processing environments dependent on said associated set of object aspects.

49. (new) The system of claim 48, devised as a network based system for enabling a user to gain access to services and files, further comprising:

a distribution and synchronization server devised for distribution and synchronization of service and file data;

a server means devised for usage by external actors to interact with services and for performing scripts defining a predetermined behavior such as data manipulation or execution of logic;

a service delivery server that is devised to act upon requests for services, loads the service, adapts it and then delivers it to a requesting user;

a file delivery server that is devised to act in response to a request service signal for a certain file, loads the file, adapts it and then delivers it to a requesting user.

50. (new) The system of claim 49, wherein the access to information is independent of the user access device, which may be a conventional computer, a WAP device, a mobile phone or any other communication device that is connectable to the system.

51. (new) The system of the claim 49, wherein the server means is used by a user to manipulate information or a service.

52. (new) The system of claim 49, wherein each of said server means , service delivery system and file delivery system has a core module that includes a service runtime device, a behave runtime device, a document format (XDF) runtime device and a service processing unit.

53. (new) The system of claim 49, wherein the system is devised to give access to different functionality dependent on the technical capabilities of the user communication device.

54. (new) The system of claim 49, wherein a service is requested by a user by sending a request service signal to the service delivery system , whereupon any possibly needed software code is executed in the server system and the service is delivered by presenting the resulting information on the communication device of the user.

55. (new) The system of claim 49, wherein the system constitutes an Internet based operating system enabling a user to gain access to services and files.

56. (new) A computer program product for processing data objects in a distributed data processing system, said distributed data processing system having a plurality of software and/or hardware nodes being communicatively connectable, comprising computer program code portions devised to direct a data processing system to perform the steps of:

defining a first object processing environment for processing objects at a first level of abstraction, said first object processing environment comprising a first object model with a first category of object aspects, wherein the processing of said first object processing environment is independent of the software/hardware platforms of said nodes;

defining a second object processing environment for processing objects at a second level of abstraction, said second object processing environment comprising a second object model with a second category of object aspects, wherein the processing of said second object processing environment is dependent on the software/hardware platform of at least one of said nodes;

synchronizing said first object model of said first object processing environment with said second object model of said second object processing environment, such that data coherency is maintained in said first and second object models;

defining an object in accordance with said first and second models and for said object associating a selectable set of object aspects from said first and second object aspect categories;

processing an instance of such an object according to said first and second object models in said first and second object processing environments dependent on said associated set of object aspects.

---